Comparing and Exploring High-Dimensional Data with Dimensionality Reduction Algorithms and Matrix Visualizations

Rene Cutura, Michaël Aupetit, Jean-Daniel Fekete, Michael Sedlmair

Comparing and Exploring High-Dimensional Data with Dimensionality Reduction Algorithms and Matrix Visualizations

Rene Cutura
Technical University of Vienna
Vienna, Austria
rene.cutura@tuwien.ac.at

Michaël Aupetit*
Qatar Computing Research Institute,
Hamad Bin Khalifa University
Doha, Qatar
maupetit@hbku.edu.qa

Jean-Daniel Fekete
Université Paris-Saclay, CNRS, Inria, LRI
Orsay, France
jean-daniel.fekete@inria.fr

Michael Sedlmair
University of Stuttgart
Stuttgart, Germany
Michael.Sedlmair@visus.uni-stuttgart.de

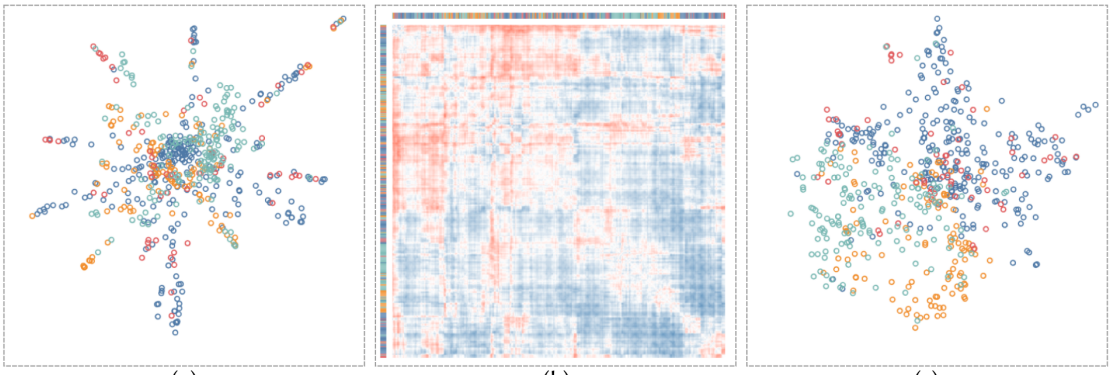


Figure 1: The scatterplots in (a) and (c) visualize the dimensionality reduced representations of two distinct subspaces of a high-dimensional dataset. The matrix visualization (b) shows the discrepancies between the distances in the two projections. The point's color in the projections encodes for data labels and serve as visual connection between them.

ABSTRACT

We propose *Compadre*, a tool for visual analysis for comparing distances of high-dimensional (HD) data and their low-dimensional projections. At the heart is a matrix visualization to represent the discrepancy between distance matrices, linked side-by-side with 2D scatterplot projections of the data. Using different examples and datasets, we illustrate how this approach fosters (1) evaluating dimensionality reduction techniques w.r.t. how well they project the HD data, (2) comparing them to each other side-by-side, and (3) evaluate important data features through subspace comparison. We also present a case study, in which we analyze IEEE VIS authors from 1990 to 2018, and gain new insights on the relationships between coauthors, citations, and keywords. The coauthors are projected as accurately with UMAP as with t-SNE but the projections show different insights. The structure of the citation subspace is very different from the coauthor subspace. The keyword subspace is noisy yet consistent among the three IEEE VIS sub-conferences.

CCS CONCEPTS

• **Human-centered computing** → **Visualization systems and tools**; *Visual analytics*; Information visualization; • **Computing methodologies** → *Dimensionality reduction and manifold learning*.

KEYWORDS

Dimensionality Reduction, Matrix Visualization, Visual Comparison, High-Dimensional Data, Subspace Analysis

ACM Reference Format:
Rene Cutura, Michaël Aupetit, Jean-Daniel Fekete, and Michael Sedlmair. 2020. Comparing and Exploring High-Dimensional Data with Dimensionality Reduction Algorithms and Matrix Visualizations. In *International Conference on Advanced Visual Interfaces (AVI '20), September 28-October 2, 2020, Salerno, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3399715.3399875

AVI '20, September 28-October 2, 2020, Salerno, Italy

ACM ISBN 978-1-4503-7535-1/20/09…$15.00
https://doi.org/10.1145/3399715.3399875

1 INTRODUCTION

This article addresses the problem of exploring and comparing high-dimensional (HD) data through multiple scatterplots coming from multidimensional projections, and a matrix visualizing the similarities and differences between them. Analyzing high-dimensional data is a core challenge in the realm of data science. One important aspect of it is to visualize these data to discover and understand patterns such as cluster structure or the importance of features/dimensions [10]. HD data are usually represented as a 2D scatterplot that shows a dimensionality reduced version of them. Popular dimensionality reduction (DR) algorithms [35] include PCA, MDS, t-SNE, and UMAP.

There exists a plethora of work on different DR methods [2, 34, 37, 38, 46, 47, 50], and on how HD data can be visualized in general [3, 25]. However, so far there has been little attention given to the role of comparison tasks in HD data analysis. Yet, comparison is a core task in the exploratory process. An analyst might for instance want to

- **(HD-2D)** compare how differences between points in HD differ from their projections in 2D,
- **(2D-2D)** compare different 2D DR projections to each other
- **(Sub-Sub)** compare subspaces with different dimensions (also known as *feature spaces*) to each other.

Similarities between data points, that analysts are looking for, are the fundamental basis for defining typical patterns in data, like density, clusters, or outliers, whether in HD space [8] or in 2D scatterplots [39], but also DR errors such as missed or false neighbors [35].

Spatialization techniques like DR techniques summarize pairwise similarities of HD data as 2D proximity between points in a scatterplot. In contrast, matrix visualizations put the data points into the columns and rows and encode pairwise similarities between these points as colored cells. Both approaches have complementary strengths and weaknesses. In a scatterplot clusters appear, for instance, as groups of points close to each other [40], while in ordered matrix visualizations they show up as blocks along the diagonal [21]. The choice of data features, similarity metric, type of DR, and parameters highly impact the resulting 2D layout summary of the data [35], while for matrices, re-ordering [6, 15] and color scale are the key factors. The main idea of our approach is to display different facets of the HD data focused on similarities, but also visualizing their discrepancies, and coordinating these complementary views with interactions to support the (HD-2D), (2D-2D), and (Sub-Sub) analytic tasks mentioned above.

To illustrate this approach, we implemented *Compadre*, a visual analysis tool for comparing distance matrices. In *Compadre* we choose to display two scatterplot DR visualizations of the data side-by-side, and add a matrix view of their similarities in-between (Figure 1). In a nutshell, the cells in the matrix show the discrepancies between the pairwise distances of the left and the right scatterplots. Blue cells of the matrix code for closer distance in the left projection and red cells in the right one, white for no difference. Compared to alternative choices, the matrix visualization offers several benefits:

- It fosters a different mental model and reading rules, specifically compared to a scatterplot usually associated with DR data directly, avoiding misleading interpretations;
- It puts the emphasis on the distances (colored cells) rather than the points;
- It introduces no error as all actual pairwise values (difference of distances) are color-coded and displayed, although it requires a good reordering algorithm to make block patterns or outliers stand out.

We explore how these interlinked visualizations can help users to discover patterns and understand features of the HD data in terms of the (HD-2D), (2D-2D), and (Sub-Sub) tasks. We also illustrate the practical relevance of our approach using a case study on analyzing the IEEE VIS paper data from 1990 to 2018 [26]. Using our approach, we can better understand which DR algorithms work better for this data. We can also analyze similarities in different subspaces, such as how coauthors compare to the keywords they use. This analysis reveals patterns such as coauthors having a different structure than cited authors, the latter being consistent with the VIS sub-conferences.

In summary, we make the following main contributions:

- We propose a framework to visually compare different dimensional spaces and sub-spaces relying through matrix visualizations and multidimensional projections,
- we illustrate this approach through *Compadre*, a visualization prototype, and
- we conduct a case study with author data of IEEE VIS publications from 1990 to 2018.

2 BACKGROUND AND RELATED WORK

We review related work on HD data analysis, the usage of matrix visualization, as well as the role of comparison tasks in visualization tools and techniques.

2.1 Visual Analysis of high-dimensional data.

Visualizing HD data is a major challenge in Visual Analytics [33]. Parallel Coordinate Plots [25] represent HD data items as polylines crossing parallel feature axes. Another alternative is to combine multiple scatterplots of axis-aligned 2D subspaces of HD data in a Scatterplot Matrix (SPLOM) and its generalization to handle nominal data (GPLOM) [22]. But HD latent structures like clusters or outliers can remain hidden along those dimensions. Interactive approaches have been proposed to discover subspaces where these patterns lie. Beyond semi-automatic approaches like ProjectionPursuit [16] and GrandTour [3], iPCA [27], InterAxis [28], or Explainers [17] use interactive Principal axes to let user explore the HD space by manipulating feature axes and items in the 2D projection. Other techniques like SIRIUS [14], IF-FI tables [49], or brushing dimensions [48] propose to support discovery of relationships between low-dimensional clusters and HD features by using dual coordinated scatterplot projections of instances and features. Tools like SeekAView [29] or Clustrophile(2) [11, 13] support interactive cluster and outlier discovery through a guided exploration of data subspaces. All the above techniques rely on linear projections of the data to maintain explainable relationships between 2D patterns and HD features.

However, some latent structures can not easily be depicted through *linear* projections. Nonlinear DR techniques like t-SNE [50] or UMAP [34] have been developed to preserve some measure of

similarity between pairs of HD items and their low-dimensional projections. As all the structural information of HD data is encoded into these pairwise similarities, these techniques are better able to represent nonlinear manifold structures revealing cluster patterns otherwise hidden by linear projections. Still, all projection techniques, linear and nonlinear, are subject to distortions which impair the analysis of HD latent structures based on 2D patterns [35]. Sleepwalk [36] addresses that issue by using two side-by-side projections of the same data linked through interactive HD similarity coloring [4] to reveal how latent structures unfold in different views. This coloring however is relative to a single item at a time which requires the user to explore all items one by one. In *Compadre*, we use a third view between the two projections. This matrix gives an interactive overview quantifying *all* the differences and similarities between the two projections. In doing so, it guides the user toward interesting patterns, draws a more trustworthy picture of the HD data, and enables the discovery of actual data traits beyond the artefacts of the projection.

2.2 Matrix Visualization

HD data can be visualized with matrices where the rows depict the n items, the columns the m dimensions, and each cell depicts a value, usually using a quantitative or diverging colormap. This technique has been used since the end of the 19^{th} century [32] and received a lot of attention in the 20^{th} century. The key issue to matrix visualization is the use of a *reordering* algorithm, sometimes called *seriation*, that selects an order for the rows and the columns so that readable and meaningful patterns appear [7]. This is possible because the content of the matrix is invariant to the row and column orders.

Visual matrices can be used to visualize data tables with color-coded values directly. They can also visualize a distance matrix or a similarity matrix of all pairs of items of HD data or the adjacency matrix of a graph, in which case they are square and usually symmetric. When the graph is directed, its adjacency matrix is not symmetric. If the graph is bipartite, its adjacency matrix is not even square and is equivalent to a data table [7].

In our work, we do not visualize high-dimensional data directly but instead differences of distance matrices. When appropriately ordered, the resulting matrix shows a visual configuration where cohesive groups and outliers are clearly visible. The distance matrix or similarity matrix is also the basic information that DR methods are trying to preserve when projecting data, so distance matrix visualization and DR visualization show different interpretations of the same data. However, they exhibit very different artifacts.

Matrix reordering methods try to group similar lines and columns together. However, the order can only arrange items in one dimension so many orders can be computed that optimize some grouping criteria at the expense of others. Still, the final matrix visualization shows all the data and does not suffer from overplotting if enough pixels are available. On the other side, DR methods use two dimensions instead of one to find an arrangement that respects HD distances in 2D, with other trade-offs regarding quality and artifacts: more degrees of freedom are available at the expenses of overplotting problems and visual artifacts such as missed and false neighbors [35].

2.3 Comparison Tasks in Visual Data Analysis

Visualization supports a variety of different tasks [1]. One very common task that underlies many visual analysis problems is comparison [18, 19]. Users might be interested in comparing entire datasets to each other, such as study data stemming from different populations, different subsets and facets of data, or the results of different data analysis algorithms. To visually encode for comparison, Gleicher et al. [19] suggest to either juxtapose visualizations, superimpose them, or to compute values that quantify differences directly and encode those. Each of these approaches has benefits and drawbacks and needs to be carefully evaluated for the problem at hand. Our work also focuses on the task of comparison, specifically of different facets of high- and low-dimensional spaces that can be depicted as distance matrices.

Close to our work are those that seek to support comparison tasks in high-dimensional data analysis in general and the process of dimensionality reduction in more detail. Currently such comparisons are most commonly performed through metrics such as stress or strain [24], which do not only allow to compare the quality of DR mappings, but also to compare different 2D projections to each other. In the latter case, usually the projection with the best metric value is chosen.

These metrics can also be fed into a visualization tool, which then allows to compare different projections. EmbComp [20] is a recent approach that seeks to bridge the usage of metrics and interactive visual analysis for analyzing different word embeddings. Another closely related approach is DimStiller [23], which guides the user in the process of comparing and selecting different DR algorithms. The main idea is to leverage workflows that simplify the process of defining and comparing different DR pipelines. Others focus less on the analysis of DR results per se, but allow the interactive exploration of different dimensional subspaces, such as the work by Tatu et al. [45].

Closest to our work is VisCoDeR [12] which uses visual parameter space analysis [40] to compare 100s to 1000s of different DR projections. An overview map allows to arrange these projections by similarity and supports tasks such as analyzing the sensitivity of DR parameter settings. Instead of focusing on comparing a large number of DR results, our main objective is to better support the piece-wise comparison of DR projections and their high-dimensional spaces and sub-spaces. In VisCoDeR, this is supported by a side-by-side view that juxtaposes a few selected DR results. This view allows to inspect projections in more detail, and the actual comparison is done by linking and brushing *selected points* across the juxtaposed projections. This approach however falls short in terms of its locality. As comparisons can only done point-by-point, it is not possible to get a quick overview over major differences, nor is it possible to spot regional and global patterns. In this paper, we show that matrix visualizations are an adequate approach to fill this gap by directly encoding the differences between *all points* of different projections and spaces.

3 GENERAL APPROACH

We now outline the general ideas behind our approach, starting with the lens through which we view HD data for our purpose, then we show three examples of analysis tasks that are supported

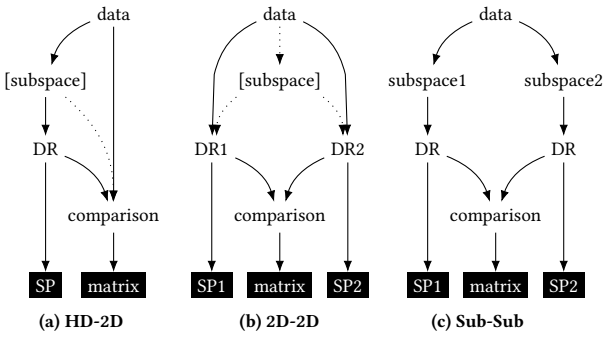


Figure 2: Visual encoding choices for the considered analysis tasks. (a) HD-2D: The HD data, or a subspace, gets projected with a DR algorithm, visualized by a scatterplot (SP), and compared with a visualization of the discrepancy matrix. (b) 2D-2D: The HD data or a subspace of it, gets projected in two different ways, each visualized as scatterplot, and compared with a visualization of the discrepancy matrix. (c) Sub-Sub: Two different subspaces of the HD data, get projected with the same DR method, each subspace projection visualized as scatterplot, and compared with a visualization of the discrepancy matrix.

with the approach, and finally we present some design alternatives that we initially considered.

3.1 Data

The main idea behind our approach is to view HD data through the lens of pairwise distances between points. Formally, we consider HD data made of n items (or cases), each of them described by an m-dimensional real vector $v \in \mathbb{R}^m$. Therefore, HD data can be represented as a $n \times m$ table D where each row i is a vector v_i. A very common approach to visualize such HD data is based on the pairwise distances between points. Formally, these distances are defined through a distance matrix $Dist(n, n)$ built from D: $Dist_{(i,j)} = \|v_i - v_j\|$. A very common approach based on that idea is the family of multi-dimensional scaling (MDS) techniques [30].

One benefit of the idea of pairwise distances is that it is flexible and applies equally well to subspaces, projections, and the HD data itself. Subspaces of an m-dimensional HD dataset are simply subsets of m' dimensions, with $m' < m$. Projections, on the other hand, usually do not use a subset of the original dimensions, but are defined instead as a linear or a non-linear combination thereof. In visualization, the most typical projections are to 2D, as it can be directly viewed by a human. While in theory 3D projections are also possible for visualization, they usually add more drawbacks than benefits [41]. Projections are most commonly created by dimensionality reduction (DR) methods. PCA [37], for example, tries to linearly project the data in such a way, that the variance of the projected points is maximized. ISOMAP [46] uses a geodesic metric to compute the shortest paths on a neighborhood graph of the HD data. It then uses MDS [47] to project the data according these geodesic distances. In doing so, ISOMAP can unfold non-linear manifolds in the HD data. Again, pairwise distances are well-defined in these low-dimensional projections and are also used by human observers for interpreting them [42].

Another important factor is the definition of the distance itself. The Euclidean distance is the most common definition, especially in 2D, but others such as the cosine distance or geodesic distance area also commonly used in HD.

Our main idea is using the pairwise differences between distance matrices of different spaces and projections as a natural way to compare them with each other. We call the result a **discrepancy matrix**. There are many ways to measure the discrepancies between two distance matrices. To start with, we can simply measure the discrepancy of two distances matrices by normalizing each distance matrix by dividing each cell by its maximum value. The normalized distance matrices of two different spaces or projections can then be subtracted from each other. We used this approach in the sequel, so all distances are normalized before comparison.

3.2 Analysis Tasks

A logical choice to visualize a discrepancy matrix is a matrix visualization. In our matrix visualization each cell encodes the discrepancy between the pairwise distances of points. We use a diverging color ramp to encode these values, e.g. from red over white to blue. Red means that two points are much closer in space A than in space B. Blue means the points are closer in B than A. White means they have a similar distance in both.

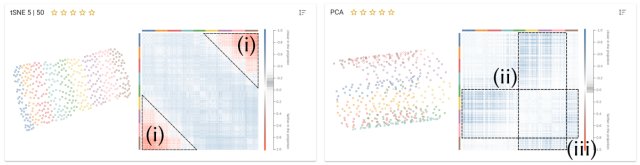
We use this visualization to support different analysis tasks see Figure 2 for an overview over these three tasks:

*(1) Analyzing DR methods (**HD-2D**).* A very common task is analyzing how well DR algorithms projects HD data to 2D space. Projecting HD data to 2D creates distortions [35]. Those distortions stem from the fact that the n-dimensional space usually cannot be mapped to correct distances in 2D (crowding problem). Hence, approximations need to be used and errors occur. Traditionally, those discrepancies are summarized into a single number such as stress or strain.

Using a matrix-based approach allows to get a much more fine-grained insight into which distances were mapped correctly versus those that were heavily distorted in the projected 2D space. Figure 3a and Figure 3b show two examples of how the matrix can be used to understand the discrepancies between HD data and 2D projections, in this case of the well-known SWISSROLL data set [44]. The points of the SWISSROLL dataset lie on a non-linear manifold that resemble a 3D swiss roll cake. We use this data as it is easy to understand for illustrative purposes.

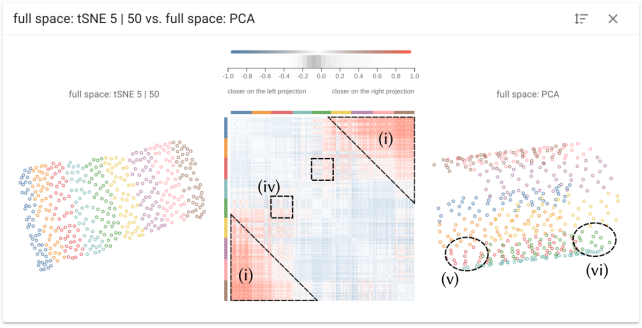
On the left of each figure, there is a 2D scatterplot projection of the data using t-SNE (3a) and PCA (3b). PCA is a linear DR method, so it is unable to unroll the manifold. In contrast, t-SNE, as a non-linear DR method, can unroll the manifold if carefully parameterized [51]. The matrix visualization on the right of the figures show the discrepancies between the actual HD SWISSROLL (in this case three-dimensional) and the two 2D projections respectively.

We can see that PCA produces lots of errors called *false neighbors*. This can be seen by the many blue cells in the matrix that indicate that points in PCA get closer than in the original HD SWISSROLL data. Two blocks 3b(ii) and 3b(iii) of blue cells appear in the matrix. The brown, pink, and purple points get squashed together (bottom



(a) t-SNE comparison to HD

(b) PCA comparison to HD



(c) t-SNE comparison to PCA

Figure 3: SWISSROLL dataset: The matrix in (a) shows the discrepancy of a t-SNE projection to its HD data, while (b) shows the discrepancy of a PCA projection on the same HD data. The manifold got unfolded by t-SNE (c) left, and ignored by PCA (c) right. The discrepancy matrix quantifies the differences between the two projections. In all cases, we use the normalized Euclidean distance to build the discrepancy matrix.

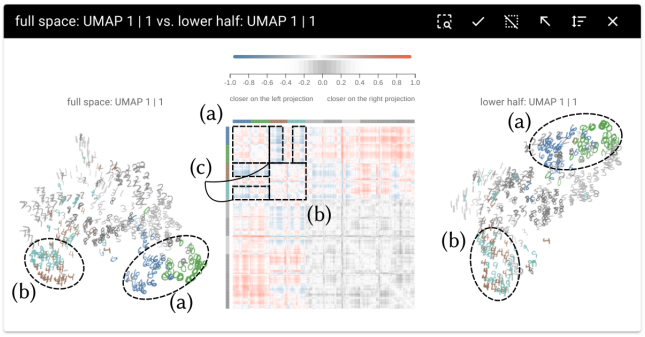


Figure 4: MNIST handwritten digit dataset: UMAP projections of the full images (left) and of the bottom half images subspace (right). The matrix is ordered by digits and uses optimal leaf-ordering within each digit. 0s and 6s (a) and 4s and 9s (b) are highlighted through interactive box selection of the top left corner of the matrix. 0s and 6s are more similar if we look at the bottom half of their image, the same for 4s and 9s, hence the red blocks (a) and (b) in the matrix. While some 4s and 9s are more different from 6s and 0s if we only look at the bottom half of the image (one almost vertical line versus half a circle) than if we look at the full image, as distances are normalized over all pixels in each subspace independently, hence the block of blue cells (c) in the matrix.

right block 3b(iii)) while green, yellow, and purple get squashed with blue, orange, and red (middle larger block 3b(ii)). While relative distances within and between adjacent colors are well preserved (large white blocks along the diagonal) except for points on the roll part tangent to the direction of the projection like cyan and green, or brown and pink (the blue blocks reach the diagonal).

In contrast, the t-SNE matrix shows red cells at the corners (3a(i)) and blue cells along the diagonal. This is an indication that some distances got larger (red) generating *missed neighbors* — on the scatterplot view, this is the case for blue and brown points for instance, which have been projected farther apart than their actual HD distance while tSNE unfolded the roll. The light blue along the diagonal indicates that the distances of pairs of adjacent points along the manifold got slightly shorter instead.

*(2) Comparing DR methods (**2D-2D**).* We can use the very same idea to directly compare two 2D projections to each other, after they have been projected from HD. Figure 3c shows a direct comparison of the two DR techniques mentioned above, t-SNE and PCA, again on the SWISSROLL dataset. In fact, the distance matrix of a t-SNE projection (3a) is very different from the distances in PCA (3b). Comparing those distance matrices can lead to interesting insights (Figure 3c).

Looking at t-SNE and PCA ignoring the matrix and pretending we ignore the original swiss roll latent structure, we could see qualitative differences, t-SNE keeping points of each color well aligned and uniformly distributed in strips across a rectangle, while PCA shows a clear mix of blue and purple, orange and yellow, and red and green data also arranged in strips but with a curvy pattern.

Adding the discrepancy matrix helps us quantifying these differences for every pair of points. In particular the blueish color along the diagonal indicates points of neighboring strips of t-SNE are closer in t-SNE than in PCA (Figure 3c), which is explained by the fact that the strips are aligned along the larger dimension of the enclosing rectangle in PCA while they are orthogonal to that direction in t-SNE (all distances being normalized). Also blue and brown points are farther apart in t-SNE than in PCA, and all other colors overlapping in PCA so being closer than in t-SNE, make the red color spread out from the top right and bottom left corners of the discrepancy matrix (3c(i)). The mixed blue and red blocks (3c(iv)) in the matrix are explained by the left and right sides of the PCA where the strips are shifted (3c(vi)) due to the oblique projection of the swiss roll, so pairs of points between overlapping and non-overlapping parts, like a green and a red point along the same edge of the swiss roll, happen to be relatively farther apart in PCA than in t-SNE (bluish color in the matrix), while many red and green points in the overlapping area of PCA are not that close to each other in t-SNE where no overlap occurs (reddish color in the matrix).

*(3) Comparing features (**Sub-Sub**).* From a distance perspective, comparing a 2D projection to another 2D projection is just a special case of comparing two subspaces of dimensions m and m' to each

other. Again, we argue that a matrix visualization can add interesting insights in that case. Also note that the dimensionality of the two subspaces does not need to be the same.

To illustrate this idea, we use an easy to understand example based on the well-known MNIST dataset [31]. MNIST is a collection of handwritten digit images of 28×28 pixels. The dimensions in the MNIST data are the pixels, a dimensional subspace would thus be a subset of pixels. For illustrative purposes, we thus cut the upper half of the digit images, keeping their lower half to form a 14×28 pixels image, flattened into a 392-dimensional vector.

We use *Compadre* to compare the two subspaces (full and bottom half images) using UMAP for the projections of the digits. We display all digits as small images at their UMAP location (Figure 4).Optimal leaf-ordering is applied to the matrix, first class-wise on the average column and row of each class, then on all columns and rows within each class separately so to preserve the class structure.

We select digits 6, 0, 4, and 9, which appear at the top left corner, with a box selection, which greys-out the other cells of the matrix and the other digits in the projection.

Considering the full images (left), 0s and 6s differ only by some extra line feature, as well as 4s and 9s. Therefore, those digits have rather small differences in pixel values, compared to say 1s and 8s for instance. Viewed in the lower half subspace (right), the 0s and 6s are even more similar, as well as the 4s and 9s, which explains the red cells in the matrix. The blue cells between some 4s and the 0s and 6s can be explained by the fact that looking only at the bottom half of the image (half space), some 4s and 9s have only one vertical bar, strongly different from the half circles of 0s and 6s, while the difference, hence the distance on UMAP projections, is smaller if we also consider the pixels of the upper half (full space).

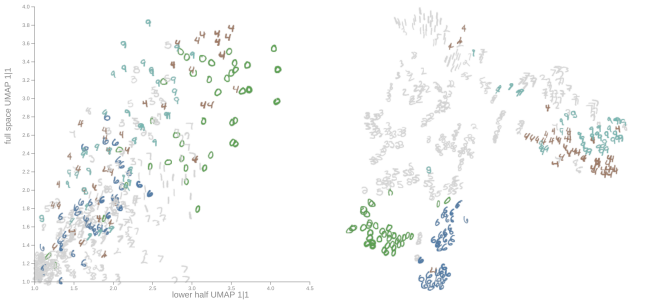
3.3 Visual Encoding

Before settling on matrix visualizations for showing discrepancies, we considered other visual encoding approaches, specifically Shepard-like diagrams and 2D projections.

3.3.1 Shepard-like Diagram. Shepard diagrams are a common approach to compare HD distances to the 2D distances in a projection [43]. We speculated that this approach might also work more generally for more arbitrary discrepancies between two sets. A Shepard diagram is a 2-dimensional plot, in which the HD distance of a pair of point gets encoded on the y-axis, and the respective 2D distance on x-axis. The diagram then shows the $n \times n$ discrepancies between the two sets of distances encoded implicitly as the distance to the diagonal of the diagram.

Showing all $n \times n$ discrepancies in a coordinate system quickly gets impracticable though. We thus aggregated all distances from one point to all other points in each space as the sum of squared distances. In that way, we reduced the number of points to draw in the diagram to n, instead of $n \times n$. We not only tried to visualize discrepancies between HD and 2D in that way, we also used it for comparison 2D to 2D (Figure 5a). Therefore, we called it Shepard-*like* diagram.

Aggregating the discrepancies makes the Shepard-like diagram difficult to read. For example in the two projections shown in Figure 4, the relation between 0s and 6s, and 4s and 9s, is not visible in the Shepard-like diagram in Figure 5a.



(a) Shepard-like diagram of aggregated discrepancies. Each image is located on each axis at the sum of its squared distances to all other images in the subspace corresponding to this axis.

(b) t-SNE projection of discrepancies. Close points are pairs which are equally close or equally far away in both UMAP projections of Figure 4.

Figure 5: Alternative visual encoding of the discrepancies of the projections in Figure 4, digits are highlighted the same way. None of these plots reveal the patterns we discovered there.

3.3.2 DR Projection of Discrepancies. A discrepancy matrix could also be simply fed to a distance-based DR algorithm to get a discrepancy projection as a scatterplot. DR algorithms such as MDS or t-SNE (used for Figure 5b) allow to input distance matrices instead of the raw data for projection. The DR algorithm then tries to maintain those distances in the projection.

Points with low discrepancy between the two scatterplot projections, should be projected nearby in the discrepancy plot, while pairs of points close in one projection and far in the other one, should be projected far apart (see Figure 5b).

As the discrepancy matrix is $n \times n$, only n points are displayed in the discrepancy plot. However, it will suffer from missed and false neighbors projection artifacts [35], making this view not trustworthy. Also, as it uses the same visual idioms as the two projections to be originally compared, it is prone to miss-interpretation. Even if there were no projection artifacts, clusters of points in the discrepancy plot are not necessarily clusters of data. Indeed, points from two distant clusters represented in the same way in both projections have all their pairwise distances equally represented, so they would appear as a single cluster in the discrepancy plot. Looking at Figure 5b, we cannot easily draw the same conclusion as for the matrix view.

3.3.3 Matrix Visualization. From a technical point of view the advantage of the matrix visualization is that each distance needs just one pixel to deliver the information of the discrepancy. With an appropriate ordering of the matrix, local (on the diagonal) and global (on the outer parts of the matrix visualization) patterns can be revealed by the viewer (Figure 4). However, the size of the matrix is the square of the number of data, hence it poses scalability issues that need to be handled for large datasets.

4 *COMPADRE*

To further illustrate these ideas, we implemented a visual analytics tool called *Compadre*[1].

4.1 Workflow

Compadre follows a top-to-bottom workflow consisting of multiple sections, which is common for interactive webpages now.

Top section: After loading a dataset, the top-most section lets users create DR projections and define subspaces. There, users can select among different implemented DR algorithms (UMAP, t-SNE, TRIMAP, MDS, ISOMAP, PCA, LLE, and LTSA), parameterize them, select a distance metric, and set seed values.

We support multi-threading for concurrently computing projections, and cache them for continuing analysis sessions. A projection listshows all cached projections.

Projections: Projections can be created on the full dataset or on defined subspaces. Each created projection then gets its own section below the top section. At the heart of each projection section is a 2D projection as scatterplot and the discrepancy matrix between the 2D projection and the HD or subspace data as matrix, supporting the **HD-2D** task (Figure 3a and 3b).

Pairwise comparisons: At the bottom is a dedicated section, which allows pairwise comparisons between the previously computed projections and subspace. This section supports the **2D-2D** and the **Sub-Sub** tasks.

4.2 Interactions

The visualization sections come with a set of interactive features, such as options for reordering the matrix and different point representations. Reordering a matrix in *Compadre* is possible with *optimal leaf-ordering* or *spectral reordering*, either globally or locally within and between classes. This supports the analysis of patterns at different scales. Examples of customizing data point representations includes thumbnail for image datasets, such as MNIST, or parallel coordinate glyphs for datasets with few dimensions.

Colorization of data points can be single hue, categorical if the dataset has labels, or by a continuous colormap to link different views together visually. All scatterplots and matrices show a tooltip when hovering over a point providing the respective name and label. A lasso or rectangle selection allows to select and view multiple items at once. Both selections support linking and brushing, that is selections are also propagated to all other views, helping to understand correlations between them.

4.3 Implementation

We implemented *Compadre* in JavaScript with the use of *vue.js*. The library *reorder.js* [15] is used to reorder matrices. For the visualizations, we use *d3.js* [9]. The implementation can be found here https://github.com/saehm/compadre.

5 CASE STUDY: VIS PUBLICATION DATA

The VISPUBDATA dataset, provided by default in *Compadre* and used in this case study, stems from the original Vis Publication dataset (VisPub) [26]. The original dataset is organized by articles whereas the VISPUBDATA dataset is organized by authors. We generated it by extracting all the authors that occur more than twice, all the distinct keywords of the original dataset that occur more than twice, and all the cited authors from the dataset when they are cited more than 80 times. Other articles are filtered out, in addition to those with no abstract or keywords. In this extracted data, we construct a vector for each entry (author) with one row per coauthor, one row per keyword, and one row per cited author. Each of these rows contains the count of occurrences of the related coauthor or keyword or cited author.

This process led to an HD dataset with 578 authors and 1490 (sparse) dimensions, which are divided in three subspaces. The first 611 dimensions define the coauthor network, the next 498 dimensions define the keyword subspace, and the last 381 dimensions define the citation network. The articles of the original VisPub dataset have been published in one of the child conferences of the IEEE VIS Conference: SciVis, InfoVis, VAST, or Vis; we also label the authors by the child conference where the author published most.

With *Compadre*, we are interested in investigating the differences between these three subspaces, as they may bring some insights into the VIS community.

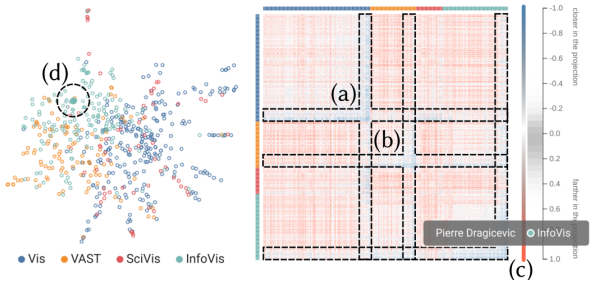


Figure 6: When loading the VISPUBDATA dataset and visualizing it with UMAP, the matrix reveals, in blue, authors who are closer in the projection than in HD. These are core authors, because UMAP brings closer heavily clustered points. Hovering over the matrix reveals the identity of these authors, here Pierre Dragicevic (c, d) who has been among the most prolific InfoVis authors in the last decade. The same pattern exists for SciVis and VAST (a, b).

Figure 6 shows an overview of the community taking into account the whole vectors projected using UMAP. The matrix shows the discrepancies between the HD and the 2D distances (**HD-2D**), the blue area are shorter distances in 2D than in HD and the red ones are longer. The points are colored according to the conference: dark blue for Vis, orange for VAST, red for SciVis, and cyan for InfoVis, and the same colors appear in the border of the matrix associated with the author of the row (respectively column). Here, the matrix has been reordered by conference using the optimal leaf-ordering algorithm [5], that seems to very effectively bring together the most prolific authors on the right side. Many patterns are immediately visible from the matrix, like the authors overlapping many conferences. SciVis at the top is very similar to Vis in red below, witnessing that the visualization conference started with

[1] https://renecutura.eu/compadre

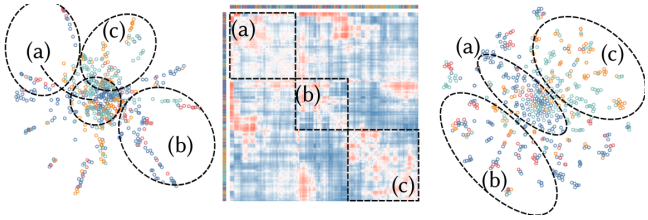


Figure 7: Using only the coauthor subspace removes the noise added by the other two subspaces and should reveal visual clusters using appropriate projections. We see that the UMAP projection varies significantly from the t-SNE projection; the matrix shows three clusters regarding the distance discrepancies: bright on the top left (a) and on the bottom right (c), and blue in the middle (b), blue points are closer in the UMAP projection than in the t-SNE projection.

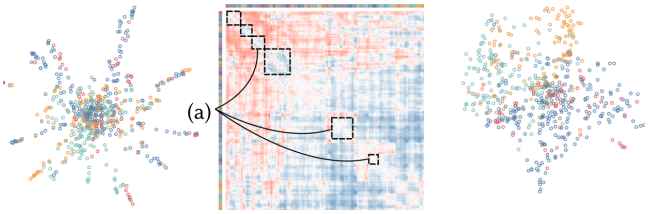


Figure 8: Comparing the coauthor (left) and keyword (right) subspaces using UMAP. The blue area on the diagonal (a) reveals a tight cluster of coauthors that work on volume visualization.

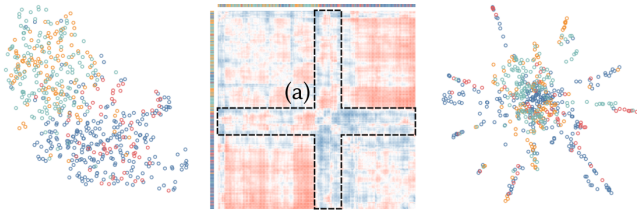


Figure 9: Comparing citations (left) with coauthors (right), we could expect the shapes to be similar if authors heavily cite their own articles. This is not the case, since the two projections have a very different shape, but the citation network is more spread than the coauthor network, while still showing a clear separation between the InfoVis/VAST papers (top left) and Vis/SciVis papers (bottom right). (a) shows a tight cluster of 75 authors that cite themselves frequently.

scientific visualization and split later. We also see the high overlap between VAST and InfoVis: the patterns are very similar in the two bands.

When hovering over the prolific authors, their projected points are highlighted in the UMAP projection. It is also insightful to see that whereas the matrix packs them all in the same area, the projection mostly spreads them all around the scatterplot.

Yet, the projection is not revealing clusters or visual patterns clearly, and knowing that keywords are typically used in a noisy fashion by authors, we wonder if the coauthor subspace is cleaner than the joint subspaces.

Figure 7 compares two 2D projections of the coauthors subspace using UMAP and t-SNE (**2D-2D**). The overall shapes are different but the information is similar since most of the clusters are depicted, either as lines for UMAP or as groups in t-SNE. UMAP seems to push points away more than t-SNE. UMAP shows spikes radiating from the center. Each spike ends with the prolific authors revealed as in the first view, but is much more visible now. Authors who collaborate with multiple communities around the center of the projections are better outlined with t-SNE (a). The discrepancy matrix shows this tendency regarding the distances except for the (b) area that is more clumpy (more red in the matrix) in t-SNE. For understanding communities, t-SNE seems more appropriate than UMAP, which reveals strong coauthor ties with its spikes.

Figure 8 shows the comparison between the coauthor (left) and the keyword (right) subspaces projected with UMAP (**Sub-Sub**). The keyword projection is more noisy and not well aligned with coauthors, indicating that the visualization topics exist independently to authors. Using our lasso tool, we can explore clusters of related authors on the left, and see the keywords they use highlighted on the right. The blue areas on the matrix diagonal reveal groups of authors and keywords that are consistent. In our dataset, *volume rendering* and *volume visualization* are consistent with a dozen of researchers who seem specialized. We cannot find other areas that are as consistent.

Figure 9 shows the citation subspace (left) compared to the coauthor subspace (right) (**Sub-Sub**). They are different in shape, the citation projection does not reveal social ties between the authors, but mostly scientific ties. The matrix does not show strong patterns on its diagonal, except at the center where about 75 authors have roughly similar distances in the two projections. This group contains the core InfoVis/VAST authors, who cite each other frequently.

6 CONCLUSION

With the advent of data science the analysis of high-dimensional data has become invaluable. In this work, we discussed how visualizing discrepancy matrices between different spaces and projections can foster insights into patterns in HD data and help validate them.

We illustrated the value of our approach with a basic set of distance and similarity definitions. Future work needs to investigate more complex definitions and setups, and might investigate them in longitudinal or controlled experiments with users. So far, we have focused on the pairwise comparison of distance matrices. A natural extension would be to use our approach to compare or summarize differences between multiple spaces or projections.

To conclude, we argue that matrix visualizations should be an important component of the data scientists toolbox, allowing for an easy-to-understand yet rich way to inspect discrepancy matrices.

ACKNOWLEDGMENTS

This work was supported by the FFG ICT of the Future program via the ViSciPub project (no. 867378).

REFERENCES

[1] Robert Amar, James Eagan, and John Stasko. 2005. Low-level components of analytic activity in information visualization. In *Proceedings of the IEEE Information Visualization Symposium.* IEEE, Piscataway, NJ, USA, 111–117.

[2] Ehsan Amid and Manfred K Warmuth. 2019. TriMap: Large-scale Dimensionality Reduction Using Triplets. arXiv:1910.00204

[3] Daniel Asimov. 1985. The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing* 6, 1 (1985), 128–143.

[4] Michael Aupetit. 2007. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing* 70, 7-9 (3 2007), 1304–1330. https://doi.org/10.1016/j.neucom.2006.11.018

[5] Ziv Bar-Joseph, David K Gifford, and Tommi S Jaakkola. 2001. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics* 17, suppl_1 (2001), S22–S29.

[6] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. 2016. Matrix Reordering Methods for Table and Network Visualization. *Computer Graphics Forum* 35, 3 (2016), 693–716. https://doi.org/10.1111/cgf.12935 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12935

[7] Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. 2016. Matrix Reordering Methods for Table and Network Visualization. *Computer Graphics Forum* 35 (2016), 24. https://doi.org/10.1111/cgf.12935

[8] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer, New York.

[9] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 17, 12 (2011), 2301–2309.

[10] Matthew Brehmer, Stephen Ingram, Jonathan Stray, and Tamara Munzner. 2014. Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 20, 12 (2014), 2271–2280.

[11] M. Cavallo and Ç. Demiralp. 2019. Clustrophile 2: Guided Visual Clustering Analysis. *TVCG* 25, 1 (Jan 2019), 267–276. https://doi.org/10.1109/TVCG.2018.2864477

[12] Rene Cutura, Stefan Holzer, Michael Aupetit, and Michael Sedlmair. 2018. VisCoDeR: A Tool for Visually Comparing Dimensionality Reduction Algorithms. In *European Symposium on Artificial Neural Networks (ESANN) (ESANN 2018 - Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning)*. i6doc.com publication, Bruges, Belgium, 105–110.

[13] Çağatay Demiralp. 2017. Clustrophile: A tool for visual clustering analysis. arXiv:1710.02173

[14] Michelle Dowling, John Wenskovitch, JT Fry, Leanna House, and Chris North. 2018. SIRIUS: Dual, symmetric, interactive dimension reductions. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 25, 1 (2018), 172–182.

[15] Jean-Daniel Fekete. 2015. Reorder.js: A javascript library to reorder tables and networks. https://github.com/jdfekete/reorder.js Accessed on 28 Jan. 2020.

[16] J. H. Friedman and J. W. Tukey. 1974. A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Trans. Comput.* C-23, 9 (Sep. 1974), 881–890. https://doi.org/10.1109/T-C.1974.224051

[17] M. Gleicher. 2013. Explainers: Expert Explorations with Crafted Projections. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 19, 12 (Dec 2013), 2042–2051. https://doi.org/10.1109/TVCG.2013.157

[18] Michael Gleicher. 2017. Considerations for visualizing comparison. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 24, 1 (2017), 413–423.

[19] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. 2011. Visual comparison for information visualization. *Information Visualization* 10, 4 (2011), 289–309. https://doi.org/10.1177/1473871611416549

[20] Florian Heimerl, Christoph Kralj, Torsten Möller, and Michael Gleicher. 2019. embComp: Visual Interactive Comparison of Vector Embeddings. arXiv:cs.HC/1911.01542

[21] Nathalie Henry and Jean-Daniel Fekete. 2006. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 12, 5 (2006), 677–684.

[22] Jean-François Im, Michael J McGuffin, and Rock Leung. 2013. GPLOM: the generalized plot matrix for visualizing multidimensional multivariate data. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 19, 12 (2013), 2606–2614.

[23] Stephen Ingram, Tamara Munzner, Veronika Irvine, Melanie Tory, Steven Bergner, and Torsten Möller. 2010. DimStiller: Workflows for dimensional analysis and reduction. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*. IEEE, Piscataway, NJ, USA, 3–10. https://doi.org/10.1109/VAST.2010.5652392

[24] Stephen Ingram, Tamara Munzner, and Marc Olano. 2008. Glimmer: Multilevel MDS on the GPU. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 15, 2 (2008), 249–261.

[25] Alfred Inselberg and Bernard Dimsdale. 1990. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization: Visualization90.* IEEE, Piscataway, NJ, USA, 361–378.

[26] Petra Isenberg, Florian Heimerl, Steffen Koch, Tobias Isenberg, Panpan Xu, Chad Stolper, Michael Sedlmair, Jian Chen, Torsten Möller, and John Stasko. 2017. vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 23, 9 (Sept. 2017), 2199–2206. https://doi.org/10.1109/TVCG.2016.2615308

[27] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, and Remco Chang. 2009. ipca: An interactive system for pca-based visual analytics. *Computer Graphics Forum* 28, 3 (2009), 767–774.

[28] Hannah Kim, Jaegul Choo, Haesun Park, and Alex Endert. 2015. Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 22, 1 (2015), 131–140.

[29] Josua Krause, Aritra Dasgupta, Jean-Daniel Fekete, and Enrico Bertini. 2016. Seekaview: An intelligent dimensionality reduction strategy for navigating high-dimensional data spaces. In *2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE, Piscataway, NJ, USA, 11–19.

[30] Joseph B Kruskal and Myron Wish. 1978. *Multidimensional scaling*. Sage, Newbury Park, california.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[32] Innar Liiv. 2010. Seriation and matrix reordering methods: An historical overview. *Statistical analysis and data mining* 3, 2 (2010), 70–91. https://doi.org/10.1002/sam.10071

[33] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci. 2017. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 23, 3 (March 2017), 1249–1268. https://doi.org/10.1109/TVCG.2016.2640960

[34] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:stat.ML/1802.03426

[35] Luis Gustavo Nonato and Michael Aupetit. 2019. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 25, 8 (Aug 2019), 2650–2673. https://doi.org/10.1109/tvcg.2018.2846735

[36] Svetlana Ovchinnikova and Simon Anders. 2019. Exploring dimension-reduced embeddings with Sleepwalk. *bioRxiv* (2019). https://doi.org/10.1101/603589 arXiv:https://www.biorxiv.org/content/early/2019/04/12/603589.full.pdf

[37] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572.

[38] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[39] A. Sarikaya and M. Gleicher. 2018. Scatterplots: Tasks, Data, and Designs. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan 2018), 402–412. https://doi.org/10.1109/TVCG.2017.2744184

[40] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. 2014. Visual parameter space analysis: A conceptual framework. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 20, 12 (2014), 2161–2170.

[41] Michael Sedlmair, Tamara Munzner, and Melanie Tory. 2013. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 19, 12 (2013), 2634–2643.

[42] Michael Sedlmair, Andrada Tatu, Tamara Munzner, and Melanie Tory. 2012. A taxonomy of visual cluster separation factors. *Computer Graphics Forum* 31, 3pt4 (2012), 1335–1344.

[43] Roger N Shepard. 1962. The analysis of proximities: Multidimensional scaling with an unknown distance function. II. *Psychometrika* 27, 3 (1962), 219–246.

[44] Dinoj Surendran. 2004. Swiss roll dataset.

[45] Andrada Tatu, Leishi Zhang, Enrico Bertini, Tobias Schreck, Daniel Keim, Sebastian Bremm, and Tatiana Von Landesberger. 2012. Clustnails: Visual analysis of subspace clusters. *Tsinghua Science and Technology* 17, 4 (2012), 419–428.

[46] Joshua B Tenenbaum, Vin De Silva, and John C Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *science* 290, 5500 (2000), 2319–2323.

[47] Warren S Torgerson. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika* 17, 4 (1952), 401–419.

[48] Cagatai Turkay, Peter Filzmoser, and Helwig Hauser. 2011. Brushing Dimensions - A Dual Visual Analysis Model for High-Dimensional Data. *IEEE Trans. Visualization & Computer Graphics (TVCG)* 17, 12 (Dec 2011), 2591–2599. https://doi.org/10.1109/TVCG.2011.178

[49] Paul van der Corput and Jarke J. van Wijk. 2016. Exploring Items and Features with IF, FI-Tables. *Computer Graphics Forum* 35, 3 (2016), 31–40. https://doi.org/10.1111/cgf.12879

[50] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

[51] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. 2016. How to Use t-SNE Effectively. *Distill* (2016). https://doi.org/10.23915/distill.00002